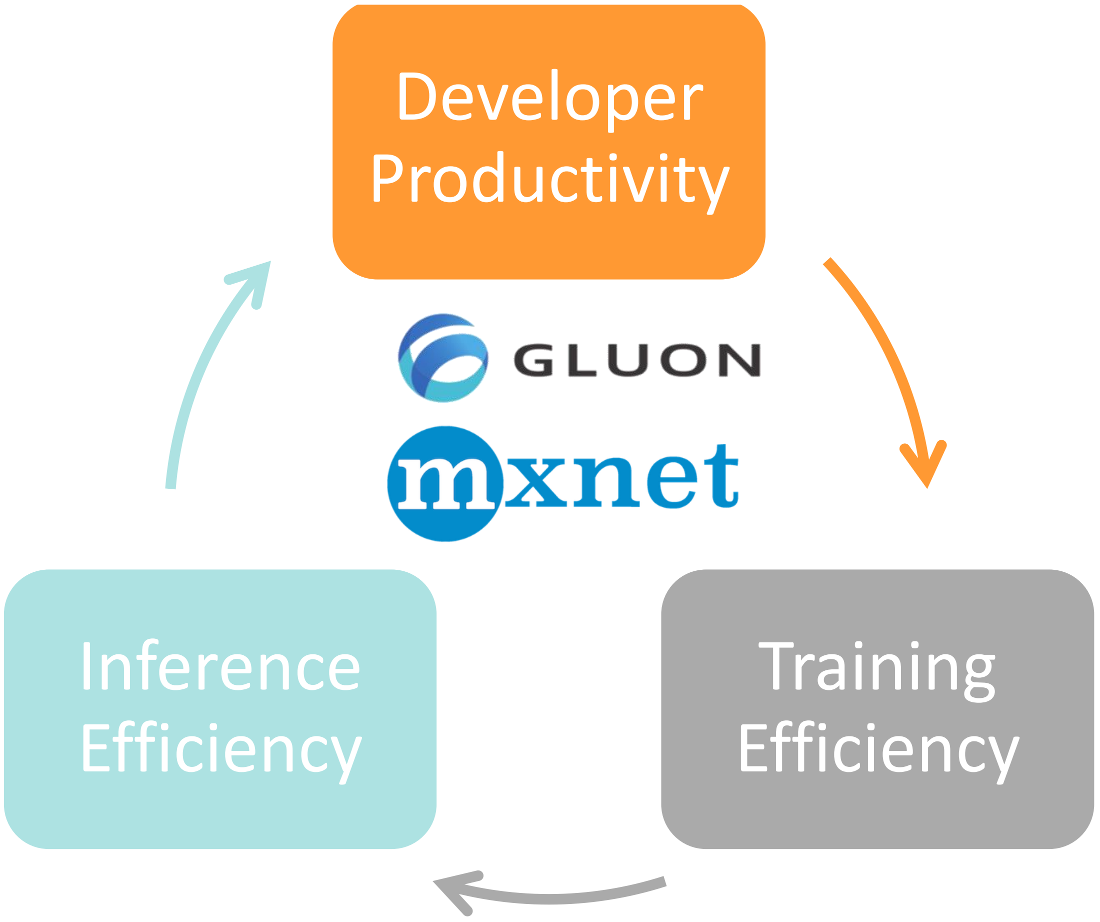
MXNET COMPUTER VISION AND NATURAL LANGUAGE PROCESSING MODELS ACCELERATED WITH NVIDIA TENSORCORES

Cyrus Vahid (Amazon), Przemyslaw Tredak (NVIDIA), 3.20.2019

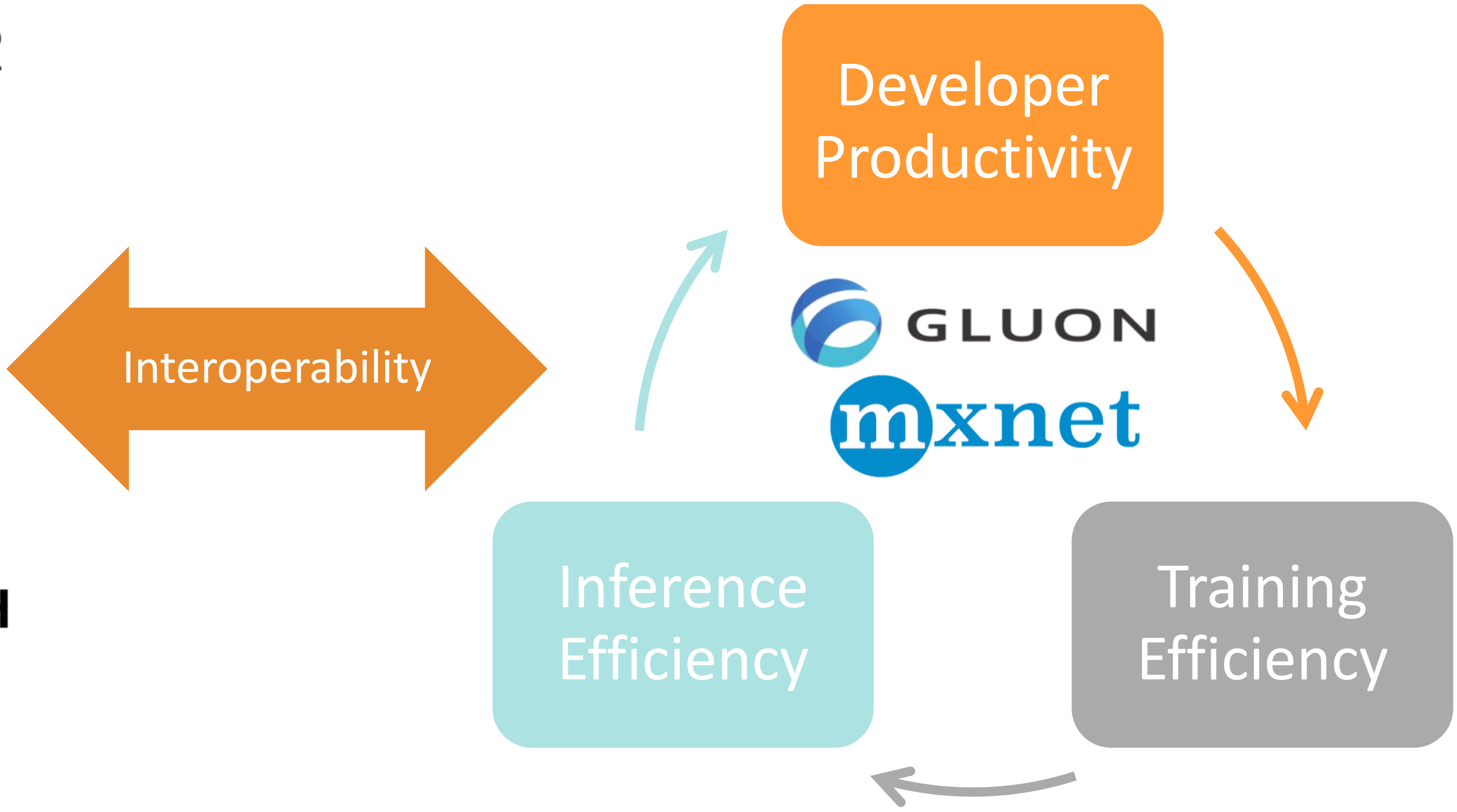
Apache MXNet (incubator)

GOALS


Developer Productivity
GLUON
mxnet
Inference Efficiency
Training Efficiency

GOALS

Caffe2
Chainer
Microsoft CNTK
Keras
PYTORCH
TensorFlow



Developer Productivity

MULTI-LANGUAGE SUPPORT

Java

Perl

Julia

Clojure

Python

Scala

C++

R

Frontend

Backend

C++

WHY GLUON

Simple, Easy-to-Understand Code

Flexible, Imperative Structure

Dynamic Graphs

High Performance

NETWORK DEFINITION IN GLUON

```
net = gluon.nn.HybridSequential()

with net.name_scope():

    net.add(gluon.nn.Dense(units=64, activation='relu'))

    net.add(gluon.nn.Dense(units=10))

softmax_cross_entropy = gluon.loss.SoftmaxCrossEntropyLoss()

net.initialize(mx.init.Xavier(magnitude=2.24), ctx=ctx, force_reinit=True)

trainer = gluon.Trainer(net.collect_params(), 'sgd', {'learning_rate': 0.02})
```

GLUON

TRAINING IN GLUON

```
for e in range(10):
    cumulative_loss = 0
    for i, (data, label) in enumerate(train_data):
        data = data.as_in_context(model_ctx).reshape((-1, 784))
        label = label.as_in_context(model_ctx)
        with autograd.record():
            output = net(data)
            loss = softmax_cross_entropy(output, label)
        loss.backward()
        trainer.step(data.shape[0])
```

HYBRIDIZE

```
net.hybridize(static_alloc=True, static_shape=True)
```

```
Wed Mar 20 16:03:29 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:00:1B.0 Off |                    0 |
| N/A   55C    P0   194W / 300W |  11371MiB / 16130MiB |     73%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:00:1C.0 Off |                    0 |
| N/A   42C    P0    54W / 300W |   5956MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000000:00:1D.0 Off |                    0 |
| N/A   42C    P0    42W / 300W |     11MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000000:00:1E.0 Off |                    0 |
| N/A   42C    P0    41W / 300W |     11MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

- NUM_GPU: 1, NUM_WORKER: 29,
- BATCH_SIZE_PER_GPU: 64.0,
- TYPECAST: <class 'numpy.float32’>
- **SYMBOLIC: False**
- **Samples/Sec: 1600**
- **epoch time: 32:00**

```
Wed Mar 20 16:09:00 2019
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 410.79       Driver Version: 410.79       CUDA Version: 10.0     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  Tesla V100-SXM2...  On   | 00000000:00:1B.0 Off |                    0 |
| N/A   43C    P0    51W / 300W |   4188MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  Tesla V100-SXM2...  On   | 00000000:00:1C.0 Off |                    0 |
| N/A   41C    P0    53W / 300W |   5956MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   2  Tesla V100-SXM2...  On   | 00000000:00:1D.0 Off |                    0 |
| N/A   41C    P0    42W / 300W |     11MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   3  Tesla V100-SXM2...  On   | 00000000:00:1E.0 Off |                    0 |
| N/A   41C    P0    41W / 300W |     11MiB / 16130MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
```

- NUM_GPU: 1, NUM_WORKER: 29,
- BATCH_SIZE_PER_GPU: 64.0,
- TYPECAST: <class 'numpy.float32’>
- **SYMBOLIC: True**
- **Samples/Sec: 3200**
- **epoch time: 16:00**

GLUON

GluonCV

GLUONCV: A DEEP LEARNING TOOLKIT FOR COMPUTER VISION

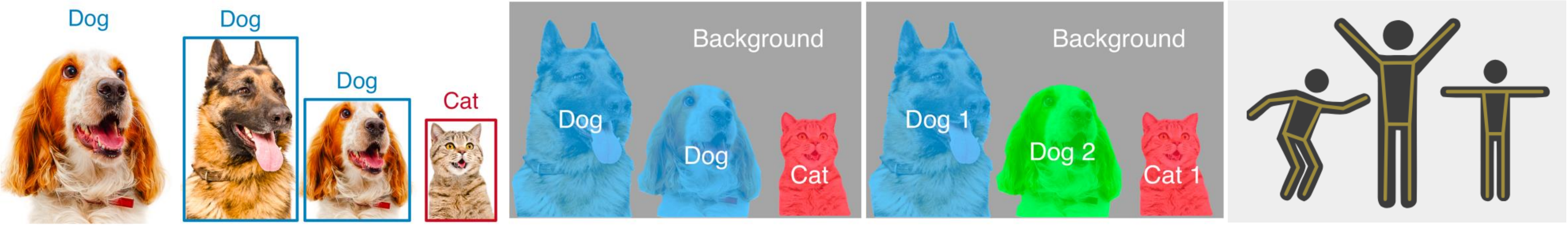


Image Classification | Object Detection | Semantic Segmentation | Instance Segmentation | Pose estimation

https://gluon-cv.mxnet.io

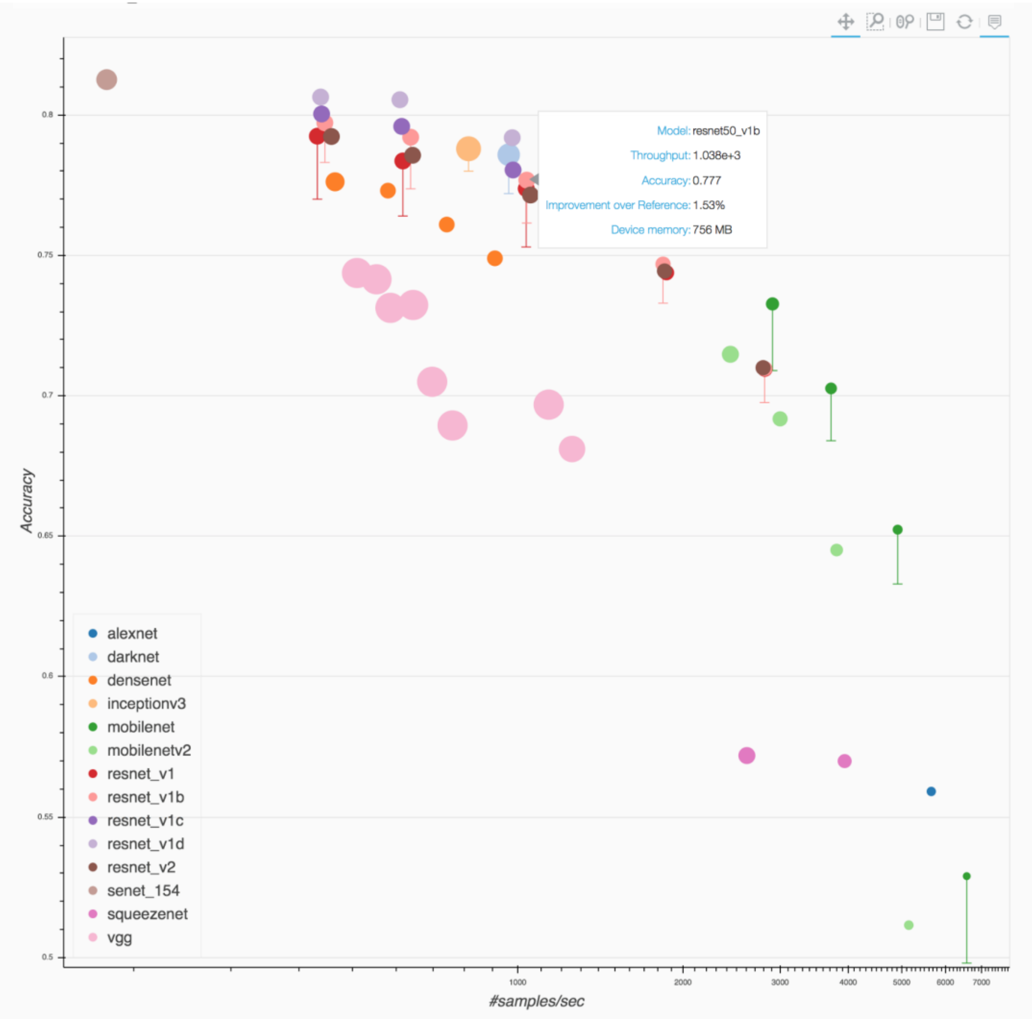
GLUONCV.MODELZOO

- 50+ Pre-trained models, with training scripts, datasets, tutorials

```
net = get_model('resnet50_v2',
                classes=100,
                pretrained=False)
```

- For a complete list of the pre-trained models please refer to: https://gluon-cv.mxnet.io/api/model_zoo.html

GLUONCV PRE-TRAINED MODELS



PRETRAINED MODELS

```
net = get_model('resnet50_v2',
                classes=100,
                pretrained=True)
```

- Transfer Learning

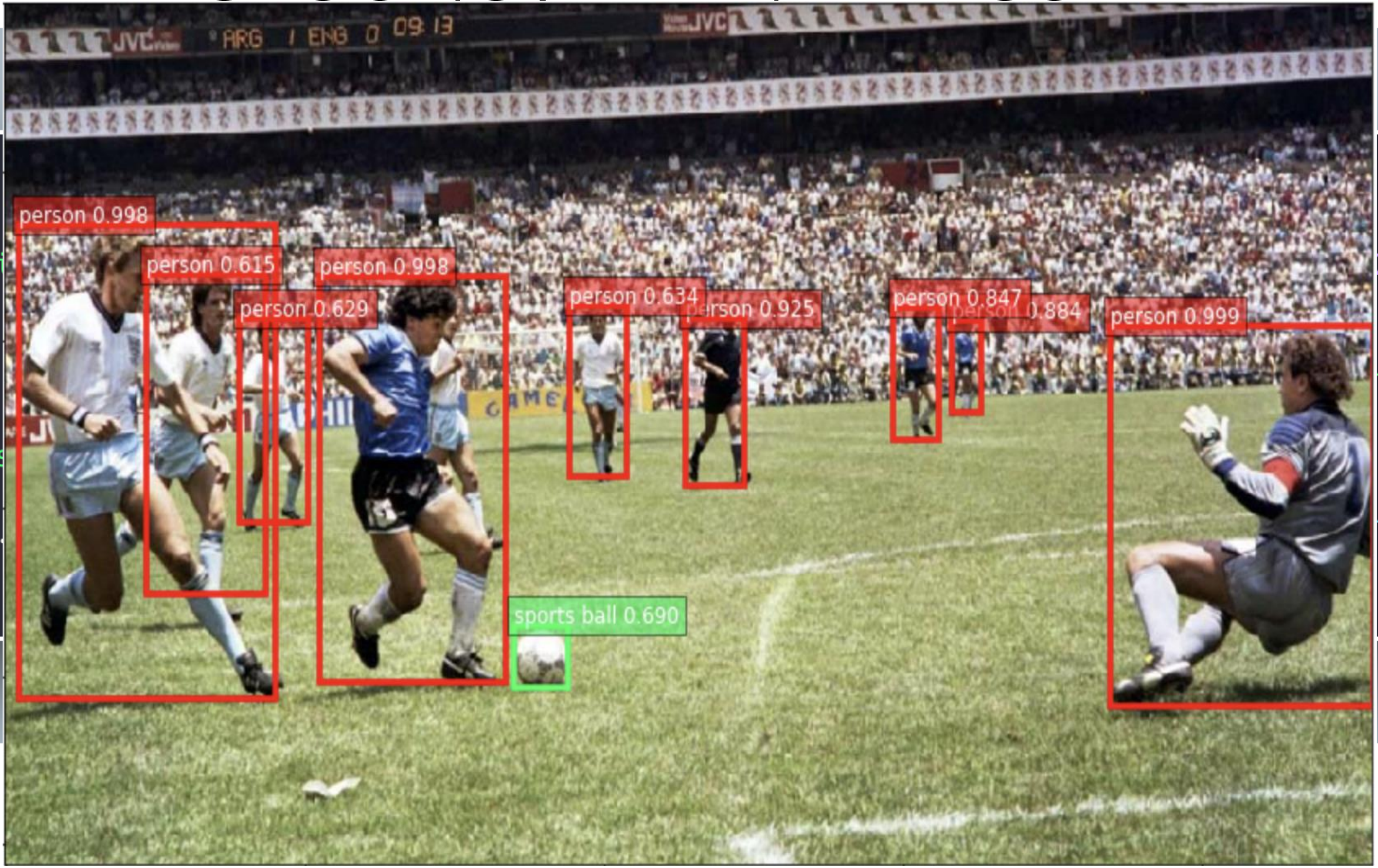
```
…pretrained=True)
```

- Inference

```
pred = net(img.expand_dims(axis=0))
ind = nd.argmax(pred, axis=1).astype('int')
nd.softmax(pred)[0][ind].asscalar()
```

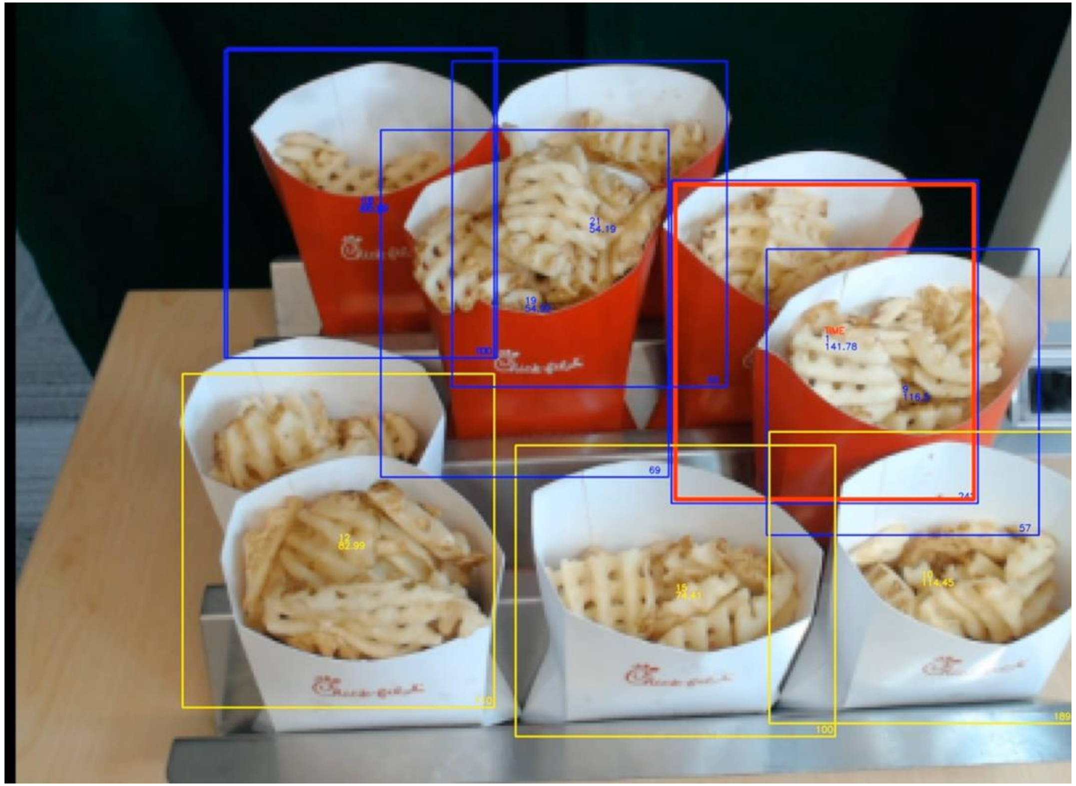
GLUONCV EXAMPLE CODE

```
x, 
ctx                                        2)

net                                        tx)

clas

viz.                                       lasses)
```



CHICK-FIL-A KEEPS WAFFLE FRIES FRESH

- Track waffle fry freshness
- Identify fries that have exceeded hold time
- Gluon Computer vision model for object detection and tracking
- A team of students with no ML expertise
- 12 months from no ML knowledge to completion



GluonNLP

FEATURES

- 300+ word embedding pre-trained models
- 5 language models
- Neural Machine Translation (Google NMT, Transformer)
- Flexible data pipeline tools
- Public datasets
- NLP examples, e.g. sentiment analysis

GLUONNLP APIS

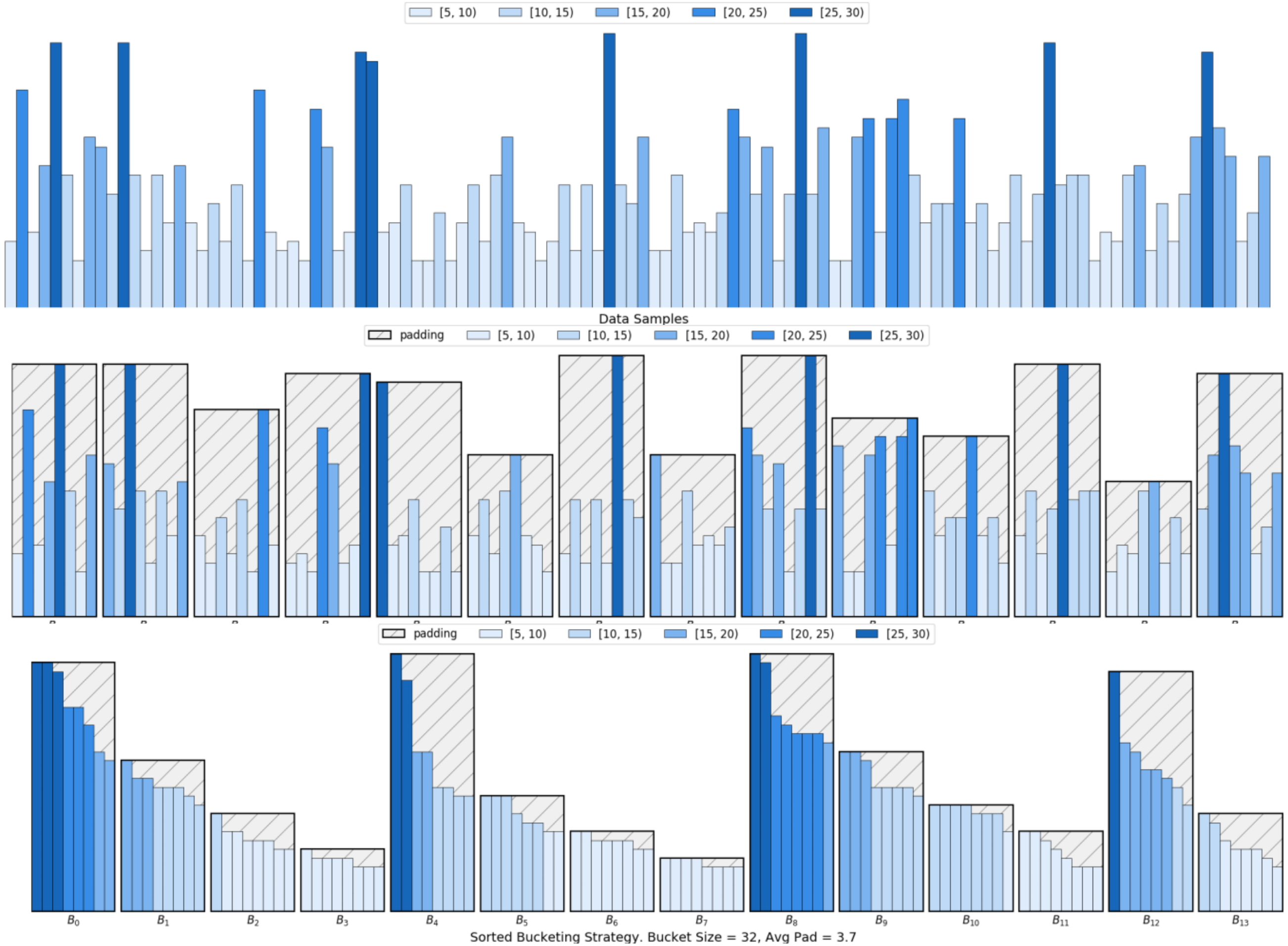
gluonnlp.data: Build efficient data pipelines for NLP tasks

gluonnlp.vocab: Provides text data numericalization and the subword functionality

gluonnlp.model:Train or load state-of-the-arts models for common NLP tasks

gluonnlp.embedding: Train or load state-of-the-arts embeddings for common NLP tasks

BUCKETING

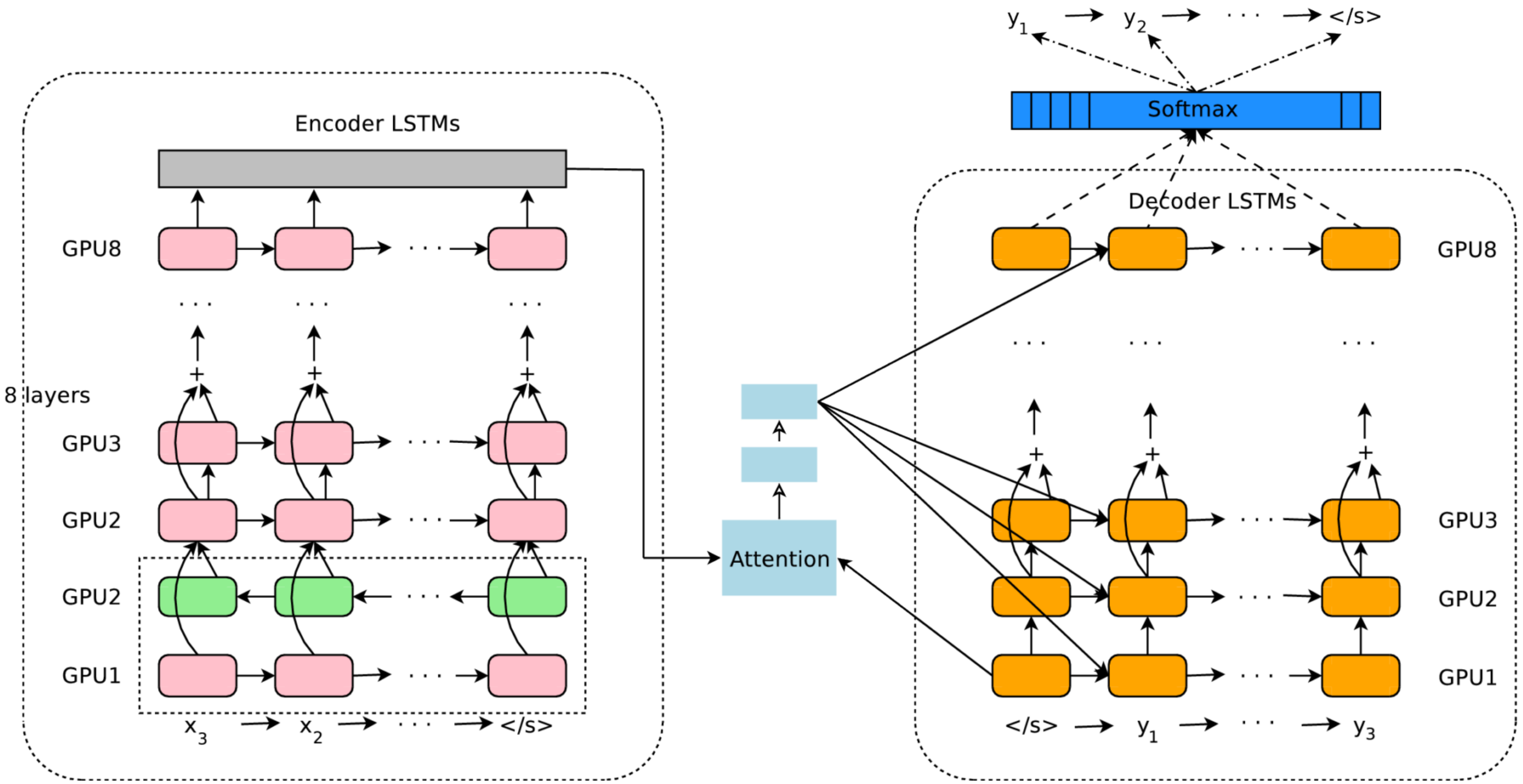


Average Padding = 11.7

Data loading
slow and memory inefficient

GluonNLP data bucketing
fast and memory efficient
Average Padding = 3.7

NMT



- Our implementation: BLEU **<u>26.22</u>** on IWSLT2015, 10 epochs, Beam Size=10
- Tensorflow/nmt: BLEU **<u>26.10</u>** on IWSLT2015, Beam Size=10

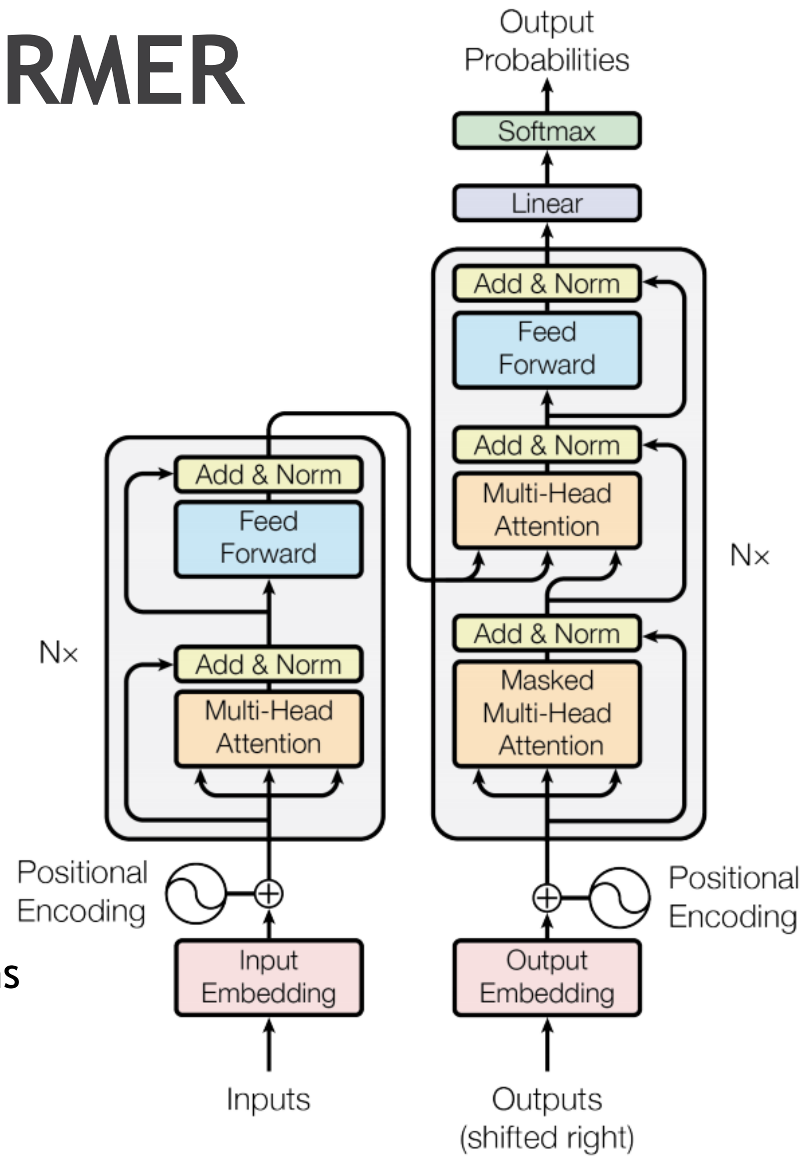
NMT - TRANSFORMER

Encoder

- 6 layers of self-attention+ffn

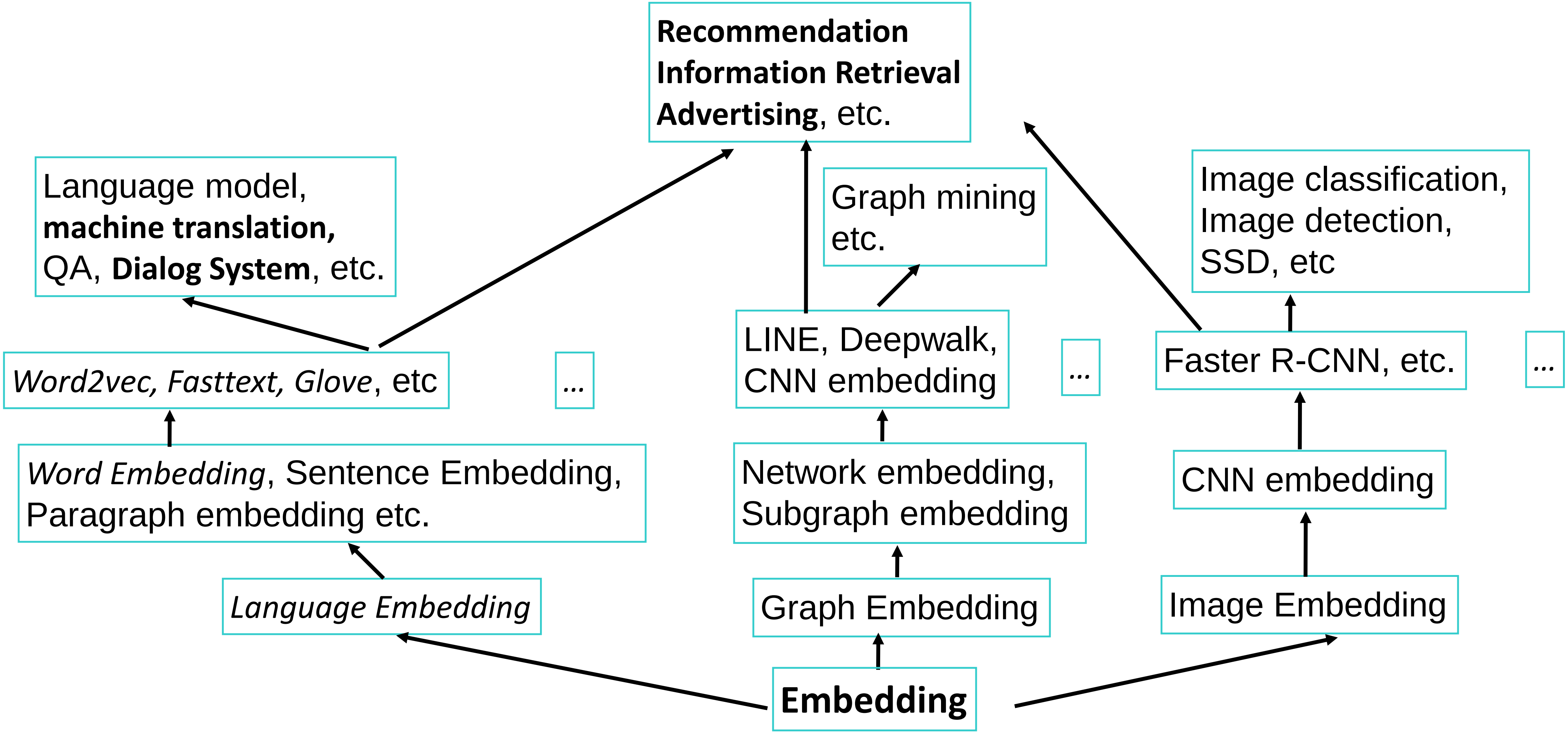
Decoder

- 6 layers of masked self-attention and
- output of encoder + ffn



- Our implementation: BLEU 26.81 on WMT2014en_de, 40 epochs
- Tensorflow/t2t: BLEU 26.55 on WMT2014en_de

EMBEDDING


Recommendation
Information Retrieval
Advertising, etc.
Language model,
machine translation,
QA, **Dialog System**, etc.
Graph mining
etc.
Image classification,
Image detection,
SSD, etc
Word2vec, Fasttext, Glove, etc
…
LINE, Deepwalk,
CNN embedding
…
Faster R-CNN, etc.
…
Word Embedding, Sentence Embedding,
Paragraph embedding etc.
Network embedding,
Subgraph embedding
CNN embedding
Language Embedding
Graph Embedding
Image Embedding
Embedding

RECAP

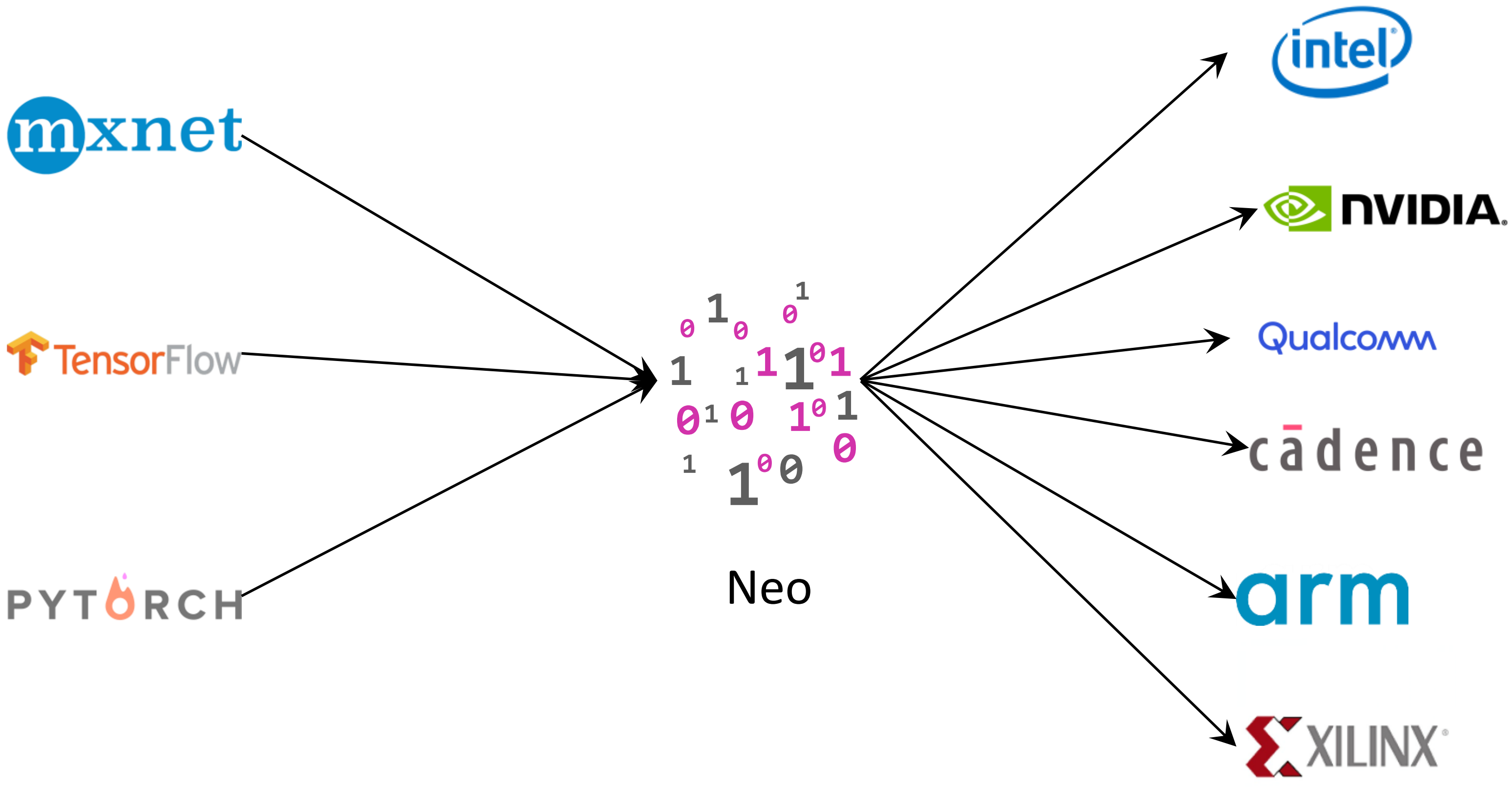
In GluonNLP, we provide

- High-level APIs
  - gluonnlp.data, gluonnlp.model, gluonnlp.embedding
- Low-Level APIs
  - gluonnlp.data.batchify, gluonnlp.model.StandardRNN

Designed for practitioners: researchers and engineers

Amazon SageMaker Neo

AMAZON SAGEMAKER NEO



TRAIN ONCE, RUN ANYWHERE WITH 2X THE PERFORMANCE

Get accuracy and performance

Automatic optimization

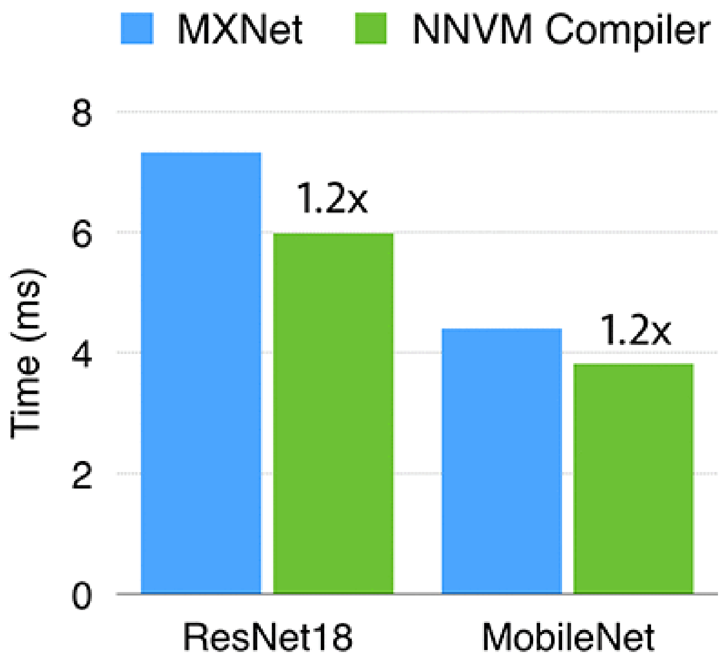
Broad framework support

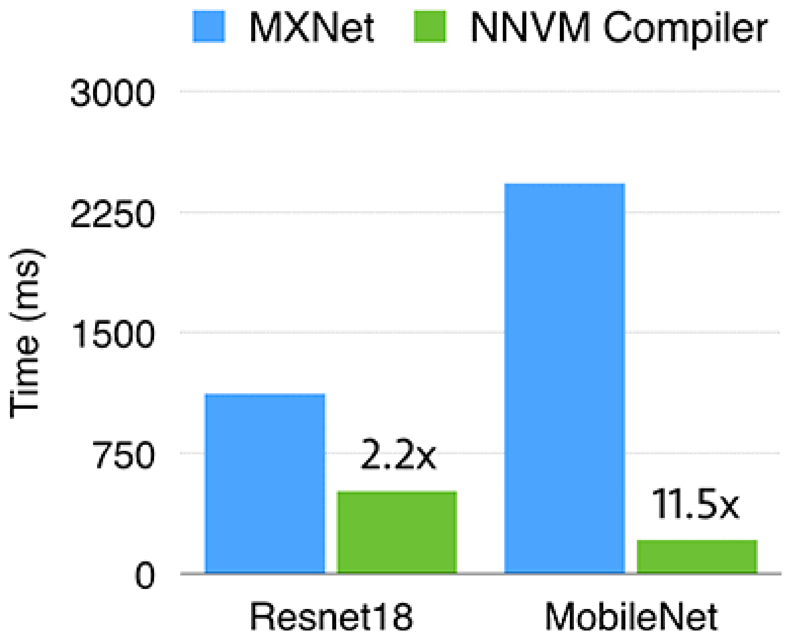
Broad hardware support

KEY FEATURES

Open-source Neo-AI device runtime and compiler under the Apache software license; 1/10th the size of original frameworks

TRAIN ONCE, RUN ANYWHERE WITH 2X THE PERFORMANCE





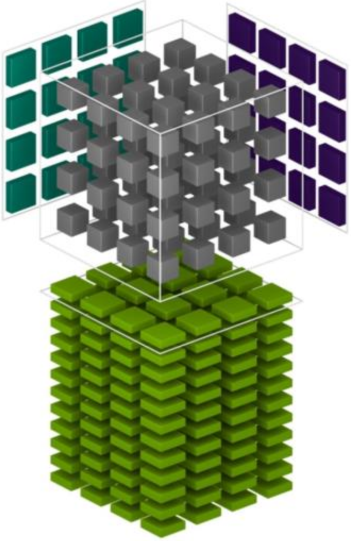
https://amzn.to/2HIj3ws

NVIDIA

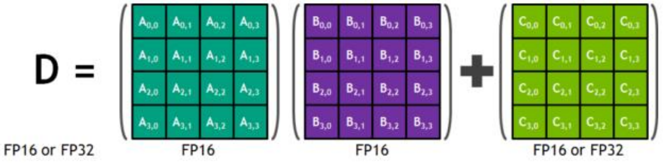
TENSORCORES AND MIXED PRECISION

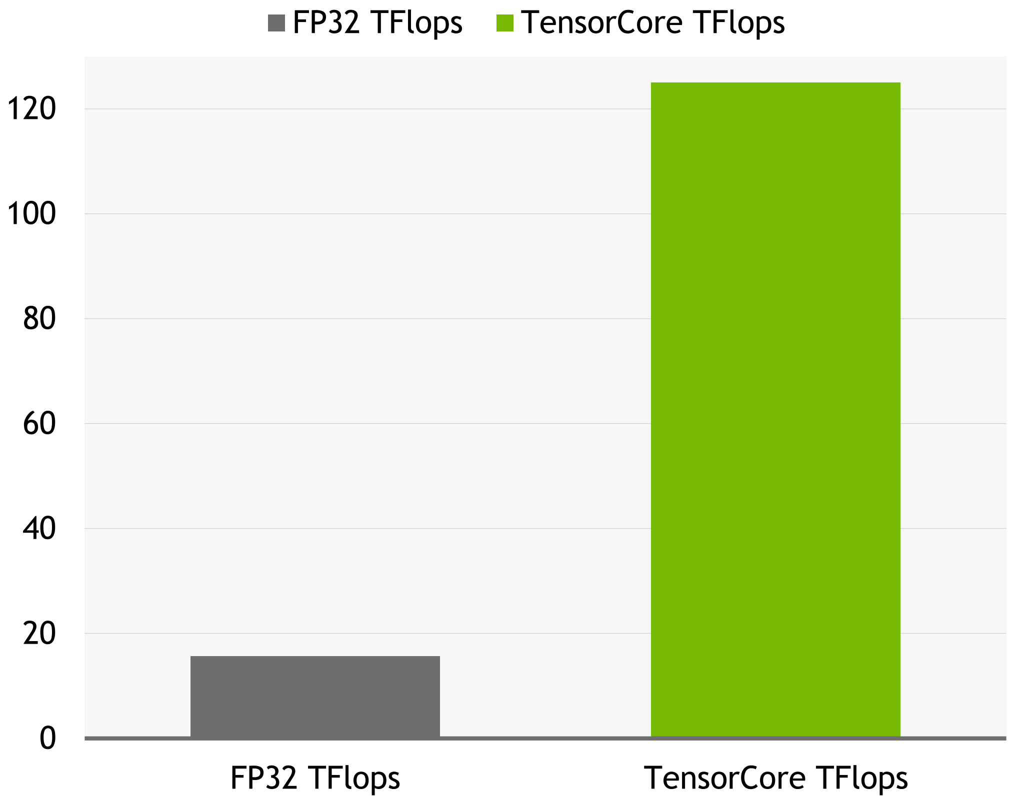
Starting with Volta, NVIDIA GPUs feature **TensorCores**

They greatly speed up matrix multiplication



Using them requires **mixed precision**





MIXED PRECISION RECIPE

Theory

- Cast input to FP16, cast back to FP32 before the softmax
- Keep "master copy" of the weights in FP32
- Scale the loss to keep gradients in FP16 dynamic range

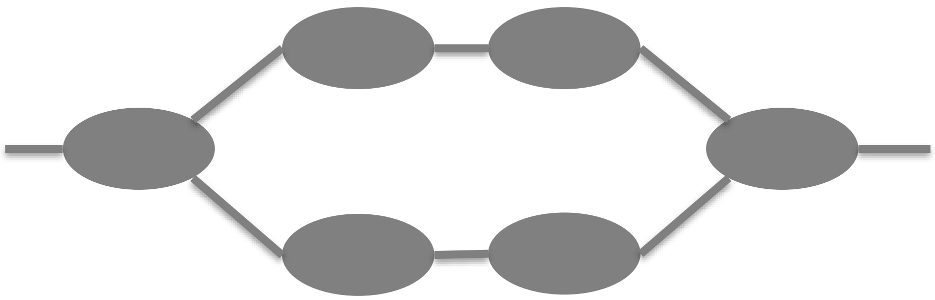
MIXED PRECISION RECIPE

In practice

- Cast input to FP16, cast back to FP32 before the softmax
  - What about Norm, Mean, etc.?
- Keep “master copy” of the weights in FP32
  - optimizer.multi_precision=True
- Scale the loss to keep gradients in FP16 dynamic range
  - What should the loss scale be? How to make it dynamic?

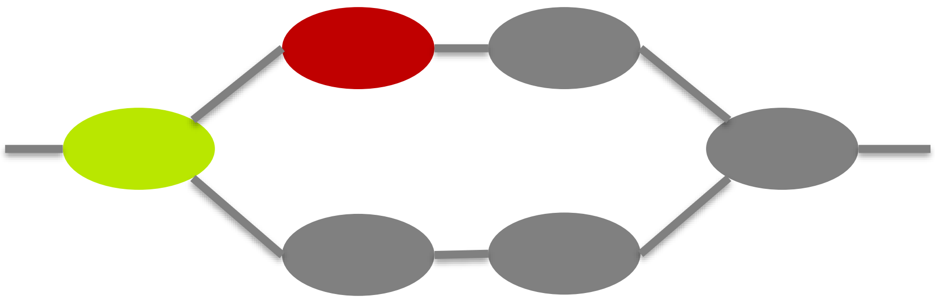
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling



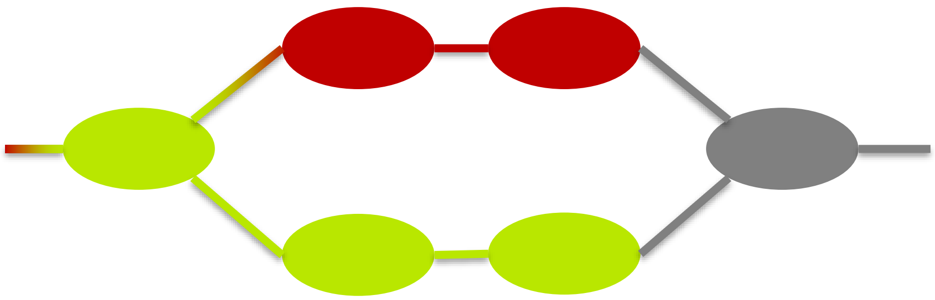
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling



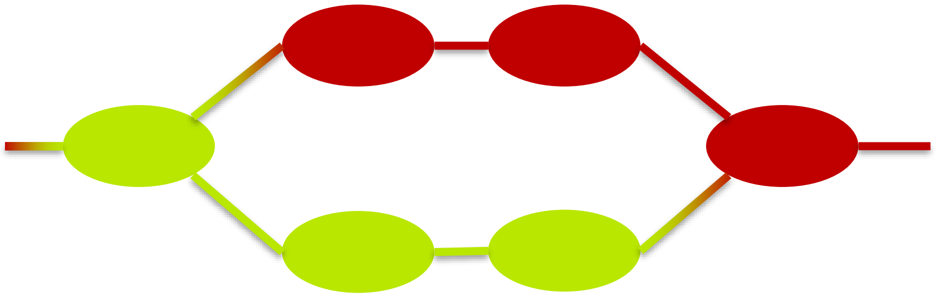
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling



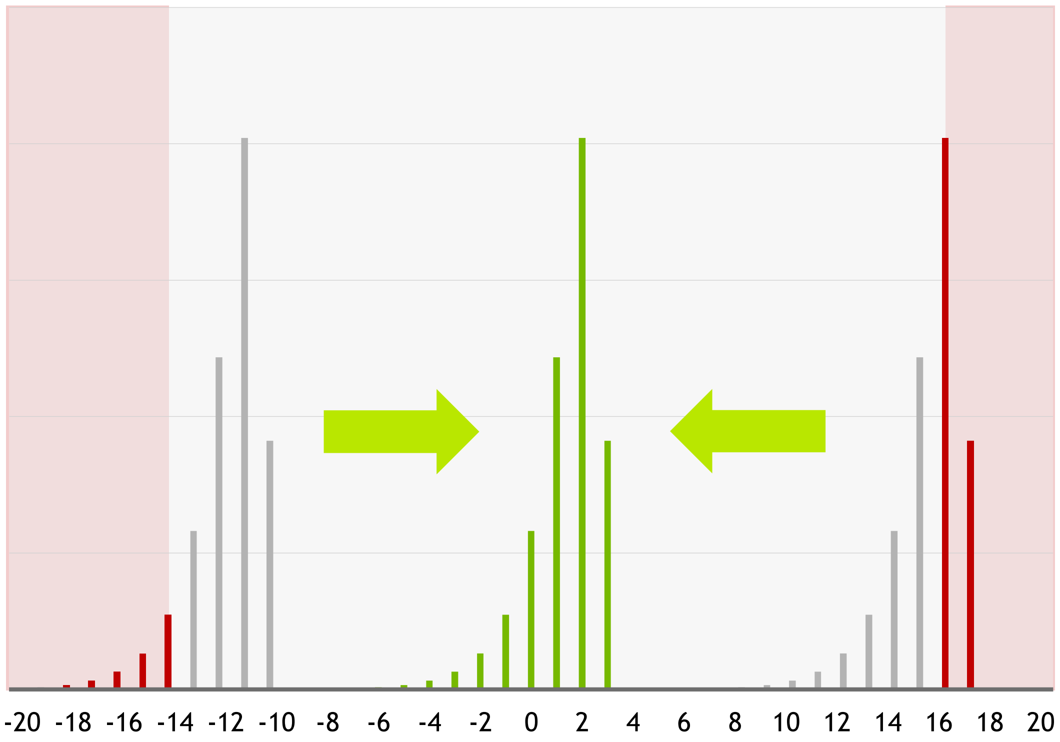
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling



AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> **FP16**
  - Norm, Mean, SoftMax, etc. -> **FP32**
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling



MIXED PRECISION RECIPE

AMP

```
net = get_network()
trainer = mx.gluon.Trainer(...)

with autograd.record(True):
    out = net(data)
    l = loss(out, label)

    autograd.backward(loss)
```

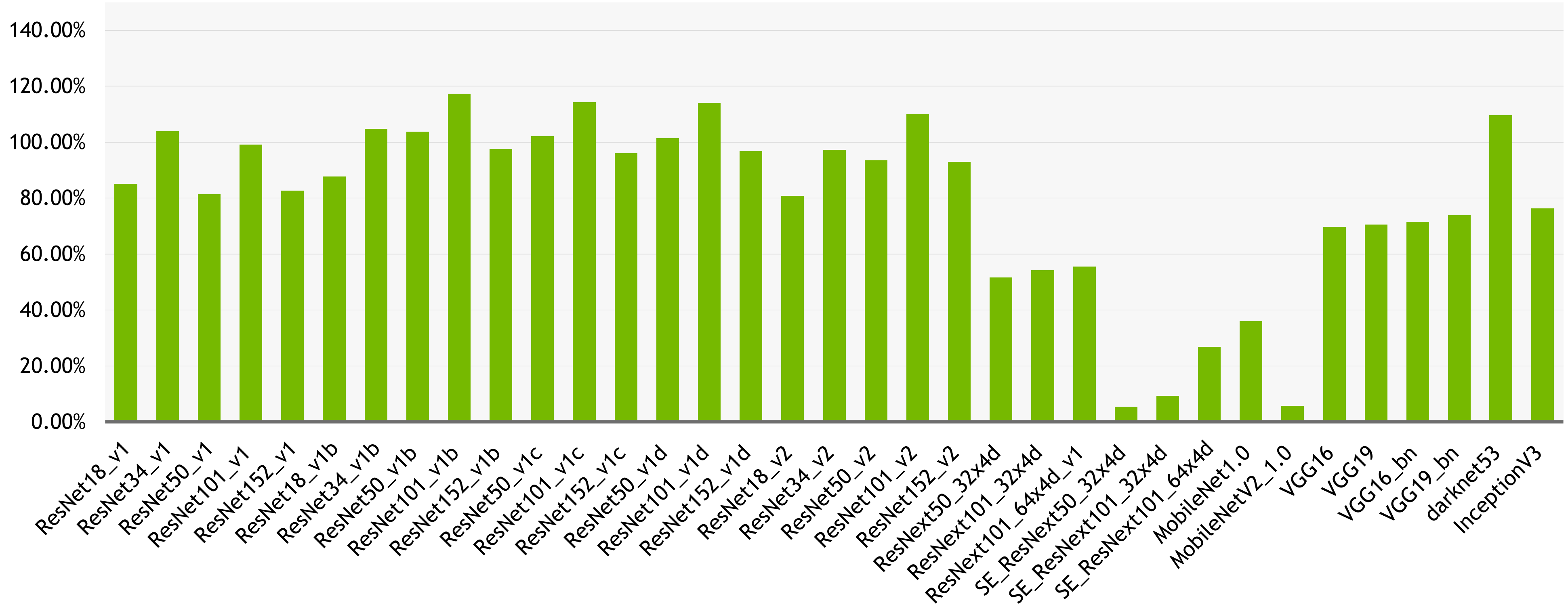
MIXED PRECISION RECIPE

AMP

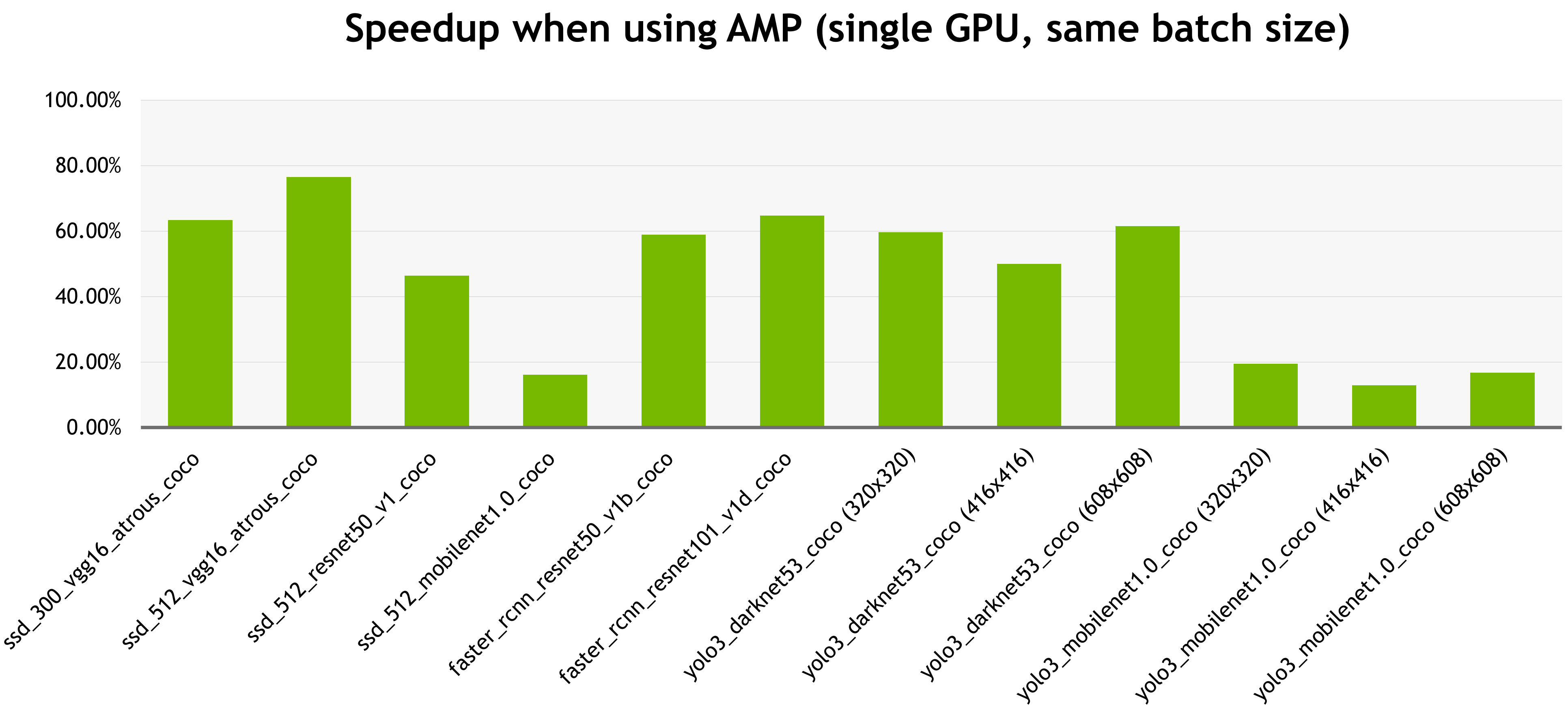
```
amp.init()

net = get_network()

trainer = mx.gluon.Trainer(...)

amp.init_trainer(trainer)

with autograd.record(True):

    out = net(data)

    l = loss(out, label)

    with amp.loss_scale(loss, trainer) as scaled_loss:

        autograd.backward(scaled_loss)
```

PERFORMANCE - CLASSIFICATION

Speedup when using AMP (single GPU, same batch size)

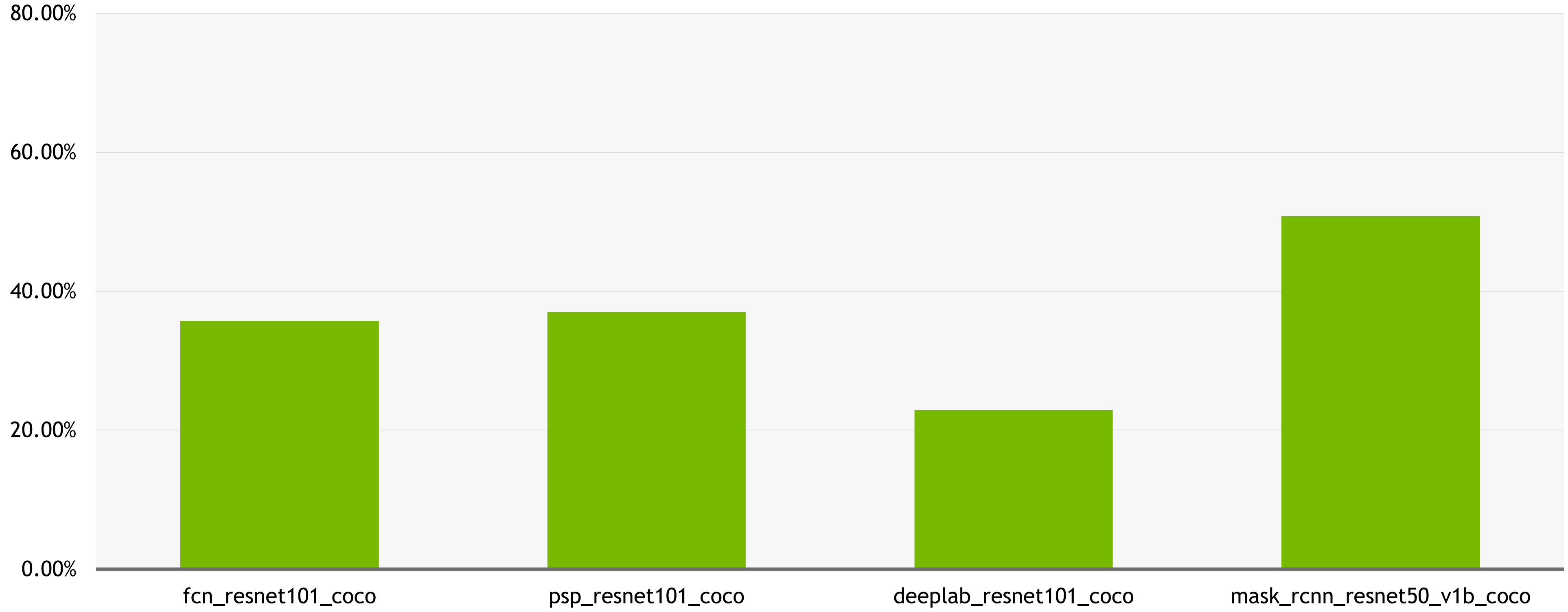


PERFORMANCE - DETECTION



PERFORMANCE - SEGMENTATION

Speedup when using AMP (single GPU, same batch size)



TRAINING OPTIMIZATION

MLPerf

MLPerf

TRAINING OPTIMIZATION

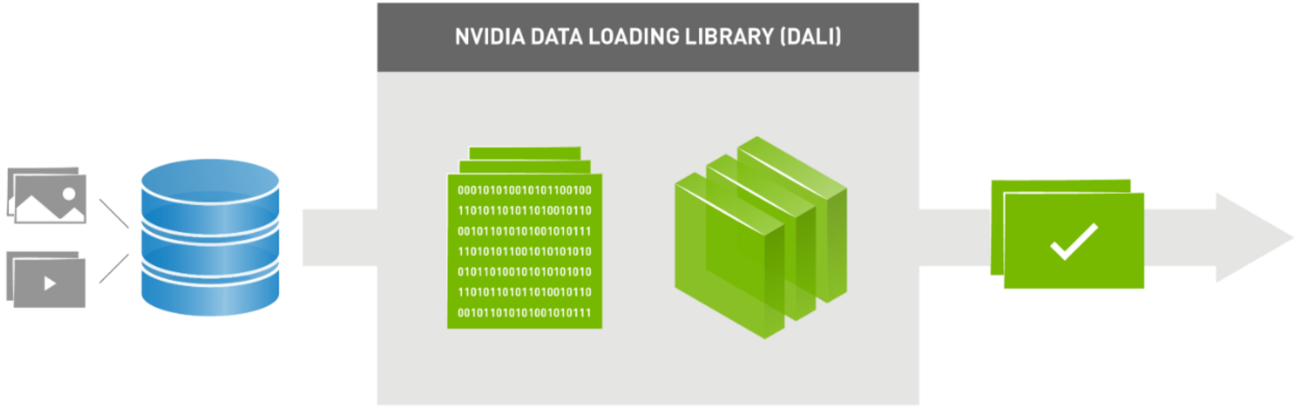
Holistic approach

- GPU kernel optimization is only one element of speeding up training

- Other potential bottlenecks
  - Data loading and augmentation
  - Operator launch overhead

TRAINING OPTIMIZATION

Data pipeline and augmentation



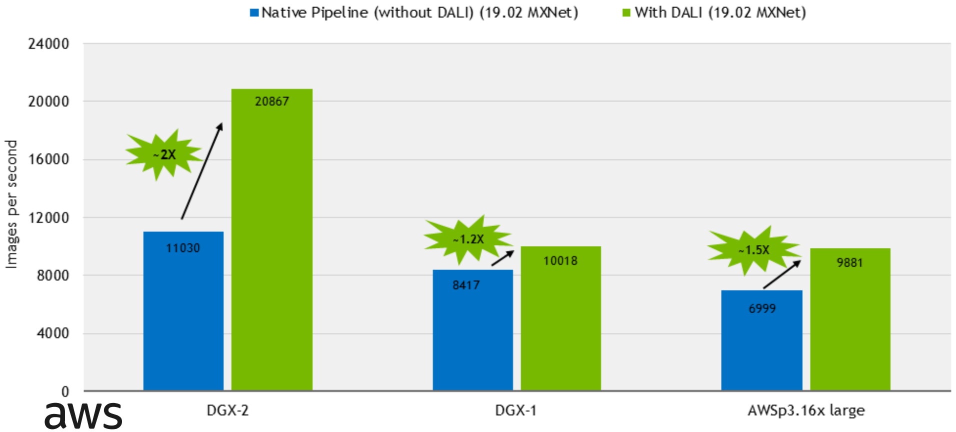
mxnet

PYTORCH

TensorFlow

...

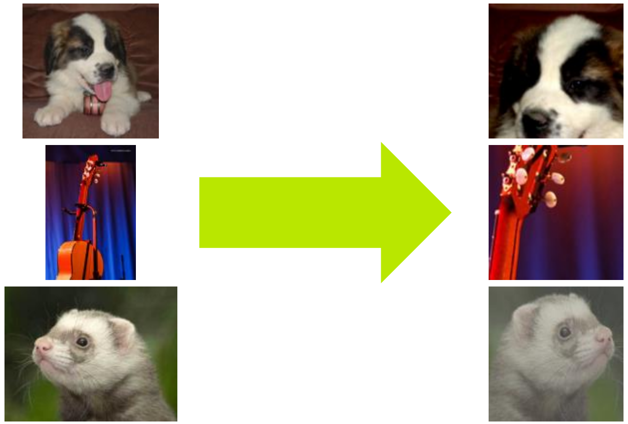
Training ResNet50 on MXNet



aws

MXNet 19.02 NGC Container, batch size: 256 for DGX-2 and DGX-1, batch size: 192 for AWSp3.16x large

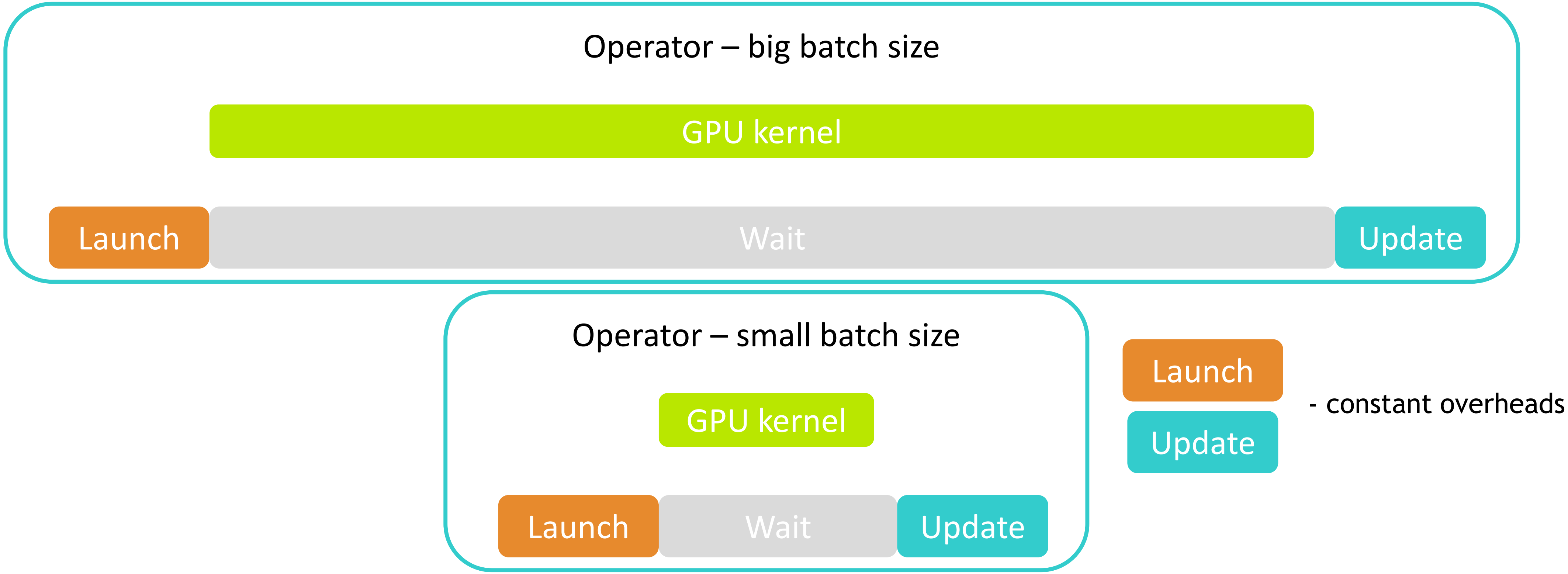
mxnet



NVIDIA

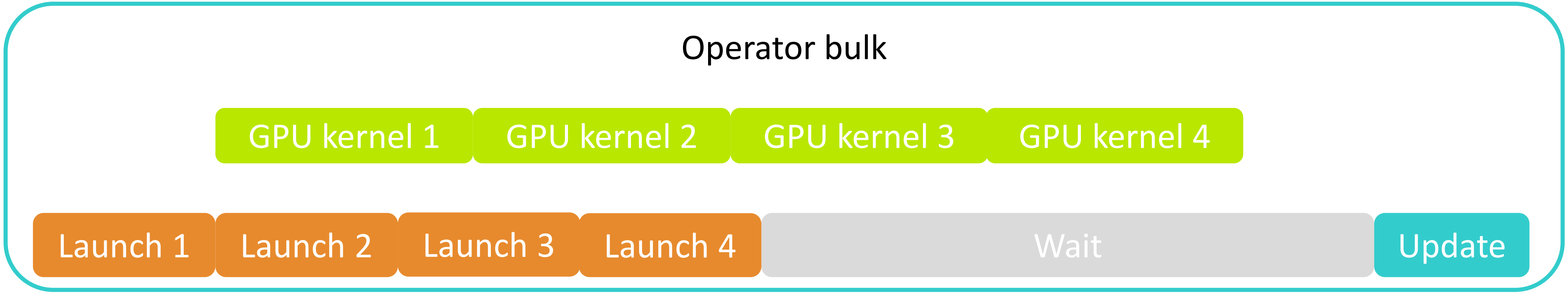
TRAINING OPTIMIZATION

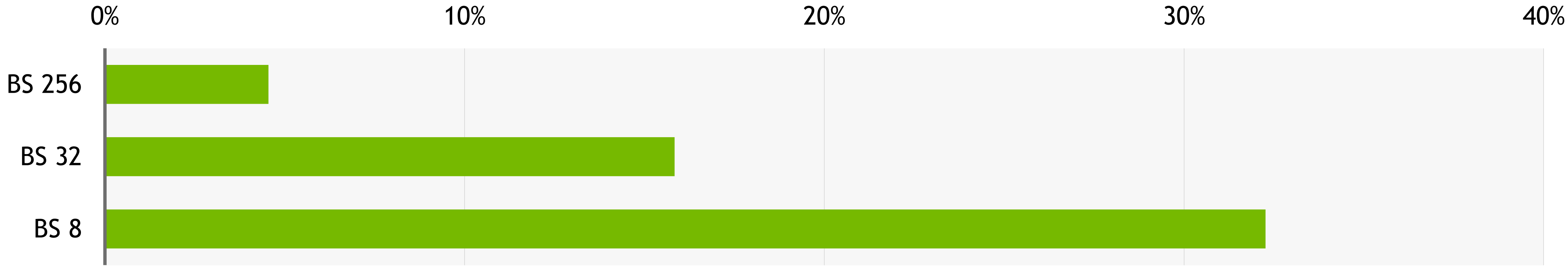
CPU optimization



TRAINING OPTIMIZATION

CPU optimization





Speedup of bulking for different batch sizes

NVIDIA

NVIDIA LED DGX SESSIONS AT GTC 2019

NOTE: For details on all DGX-related sessions, visit: GTC site and search for “DGX” or look-up session ID

NVIDIA LED SESSIONS			
Session #, Date/Time	**Location**	**Session Name**	**Product Featured**
S9483 Mon 3/18, 9am	Marriott Hotel Ballroom 3	**Creating AI Workgroups Within The Enterprise: Developers Share Their Best Practices** - Markus Weber and Michael Balint (DGX) + Customer Subtle Medical	
S9120 Tue 3/19, 10am	Convention Center Room 212B	**How to Accelerate and Scale A.I. Deployment with Proven Architecture Designs** - Charlie Boyle (DGX Product Management) and Ludwig Gamache (Head of IT ElementAI)	 
S9121 Tue 3/19, 2pm	Marriott Hotel Ballroom 3	**Deep Learning Implementers Panel: Experts Discuss The Keys to Their Success** - Tony Paikeday (DGX), Zachary Hanif (Capital One), Enhao Gong (Subtle Medical), Norman Mueller (BMW)	
S9500 Wed 3/20, 9am	Convention Center Room 212B	**Latest Deep Learning Framework Container Optimizations** - Michael O’Connor and Joey Conway (Deep Learning Software)	 
S9334 Wed 3/20, 10am	Convention Center Room 212B	**AI Infrastructure: Lessons Learned from NVIDIA DGX POD** - Darrin Johnson + Jacci Cenci (DGX Tech Marketing), Andrew Bull + Sumit Kumar (Solution Architects)	
S9893 Wed 3/20, 1pm	Convention Center Room 212B	**KVM GPU Virtual Machines: Maximizing Performance and Utilization on DGX-2** - Anish Gupta, Software Engineer	
S9241 Wed 3/20, 1pm	Convention Center Room 220C	**All You Need to Know about Programming NVIDIA's DGX-2** - Lars Nyland and Stephen Jones, Software Engineers	
S91003 Wed 3/20, 2pm	Convention Center Room 210A	**MXNet Computer Vision and Natural Language Processing Models Accelerated with NVIDA TensorCores** - Przemyslaw Tredak (DevTech Engineer) and Cyrus Vahid (Principle Evangelist AWS Deep Engine)	

DGX CUSTOMER/PARTNER LED SESSIONS @GTC

CUSTOMER/PARTNER LED SESSIONS			
Session #, Date/Time	**Location**	**Session Name**	**Product Featured**
S9292 Mon 3/18, 10am	SJ Convention Center Room 212B	**Red Hat and the NVIDIA DGX: Tried, Tested, Trusted** - Jeremy Eder, Software Engineer, Red Hat and Charlie Boyle (DGX Product Management)	
S9164 Tue 3/19, 9am	Hilton Hotel Market Room	**Advanced Weather Information Recall with DGX-2** - Tomohiro Ishibashi, Director, Weather News and Shigehisa Omatsu, CEO, dAIgnosis,Inc.	
S9983 Tue 3/19, 9am	Marriott Hotel Ballroom 5	**Edge to Core: A Meta Study of Data Complexity in AI** - James Coomer, Senior Vice President Products, DDN	
S9325 Tue 3/19, 10am	SJ Convention Center Room 220B	**Machine Learning in Action within a Large Regional Healthcare System** - Brandon Fornwalt, Associate Professor, Geisinger	
S9373 Tue 3/19, 3pm	Marriott Hotel Ballroom 2	**TPC-H Benchmark on DGX-2: A New Paradigm for OLAP and Decision Support** - Richard Heyns, CEO, and Piotr Kowalski, Senior Engineer, Brytlyt	
S9417 Wed 3/20, 3pm	SJ Convention Center Room 211B	**Molecular Generative VAEs: Parallelization, Optimization, and Latent Space Analysis on DGX-1** - Ellen Du and Joey Storer, Research Scientists, Dow Chemical Company	
S9469 Wed 3/20, 4pm	SJ Convention Center Room 231	**MATLAB and NVIDIA Docker: A Complete AI Solution, Where You Need It, in an Instant** - Jos Martin and Joss Knight, Engineering, MathWorks	
S9892 Wed 3/20, 4pm	SJ Convention Center Room 220A	**Deep Learning for Autonomous Driving at BMW** - Alexander Frickenstein, PhD Candidate, BMW	
S9406 Thu 3/21, 3pm	SJ Convention Center Room 212B	**Hybrid Cloud for Flexible GPU Resource Planning and Orchestration** - Jeongkyu Shin, CEO and Joongi Kim, CTO, Lablup, Inc.	